

eZ components

The enterprise development platform for PHP 5

20.05.2006, phpDay, Bari, Italy

Tobias Schlitt <ts@ez.no>

About me

- Tobias Schlitt <ts@ez.no>
 - IT specialist, currently studying computer science
 - Working for eZ systems on the eZ components
 - Member of the PEAR project
 - PHP Usergroup Dortmund (Germany)
-
- <http://ez.no>
 - <http://schlitt.info>
 - <http://phpugdo.de>

Agenda

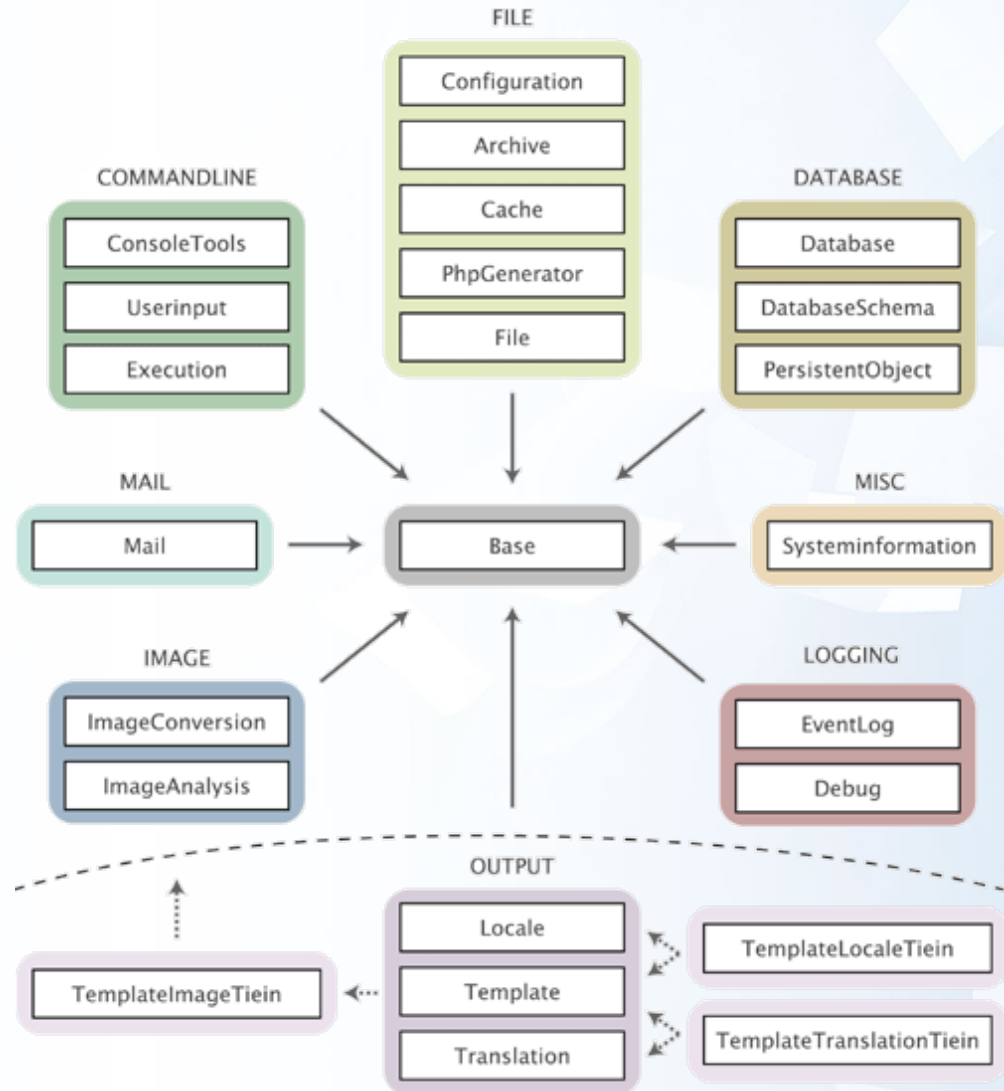
- Introduction
- Architecture / design (very short)
- Example application
- Open part

Introduction – What is it?

- A “general purpose enterprise ready PHP platform”.
- eZ components is not a framework...
- ... it is a library of loosely coupled building blocks
- Not only dedicated to web development...
- ... but focused on it.

Introduction - Overview

Share your information



Introduction – Current status

- Version 1.0.1 – stable, Version 1.1beta2 – beta
- Currently 16 components included
- More to come for 1.1
- New BSD license (completely free)
- Professional support available
- Will be the new basis of eZ publish

- New components in 1.1
 - Template
 - SystemInformation
 - File
 - DatabaseSchema
 - PersistentObjectDatabaseSchemaTiein
 - (more to come...)

Introduction – Library goals

- Provide a solid platform for PHP application development
- Clean and simple API
- Keep backwards compatibility for longer periods of time
- Stable and few regressions
- Clean IP, Open Source friendly

Introduction – Development goals

- Move to PHP 5
- Thoroughly plan the product to get a top notch API for the library
- Thoroughly tested. All code should be covered by unit tests PRIOR to implementation.
- Make sure we keep the product open enough for future development.
- Do proper documentation during the development

Introduction – Installation and setup

- Using PEAR (easy!)
 - `$ pear channel-discover components.ez.no`
 - `$ pear install ezc/eZComponents`
 - `($ pear upgrade ezc/eZComponents-beta)`
- Installing by hand (easy, too!)
 - `$ wget http://ez.no/downloads/ezcomponents-1.0.tar.bz2`
 - `$ tar -xjf ezcomponents-1.0.tar.bz2`
- Autoload code:

```
<?php
    ini_set( 'include_path', '/your/path/to/ezc:' );
    require 'Base/base.php';

    function __autoload( $className )
    {
        ezcBase::autoload( $className );
    }
?>
```

Architecture – Class names

- Pre-fixed: for namespacing
 - Use prefix ezc, because PHP has no namespaces.
- Readable: for sanity
 - Usage of descriptive class names instead of cryptic.
- Slightly Mangled: for clarity
 - ezcMail vs. Mail
 - ezcTestSuite vs. PHPUnit2_Framework_TestSuite
 - ezcMailSmtptTransport vs. ezcMailSMTPTransport

Architecture – Autoload arrays

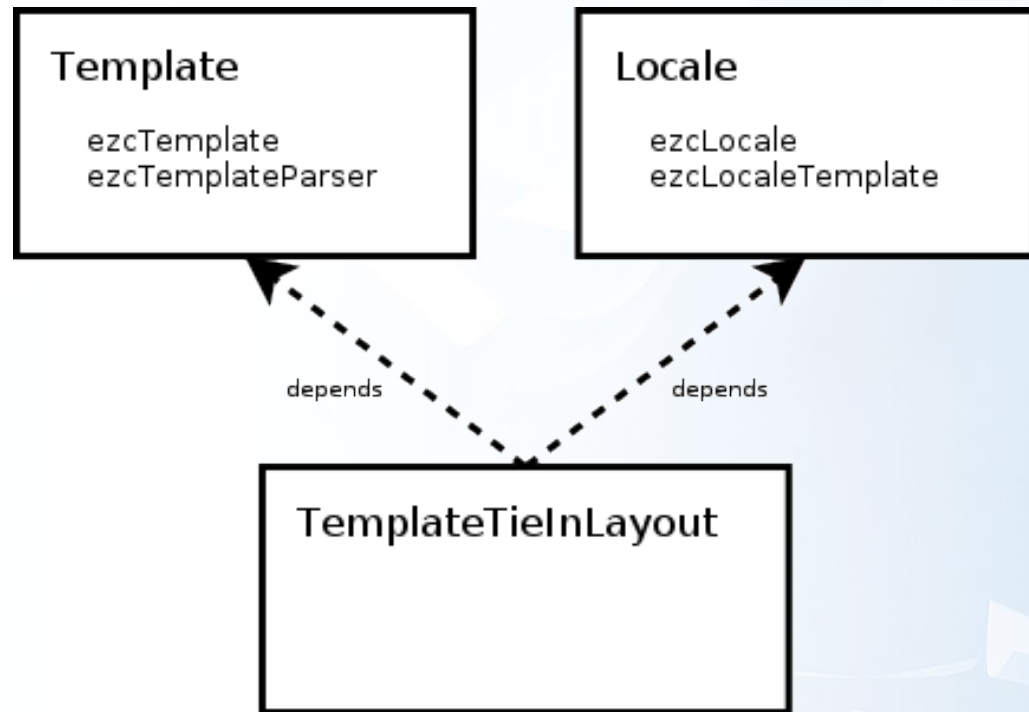
- Using the classname's part as path elements makes ugly paths:
 - `ezcMailTransportMta` ==> `mail/transport/mta.php`
 - `ezcMailTransportSmtplib` ==> `mail/transport/smtplib.php`
 - `ezcMailException` ==> `mail/exception.php`
- More logical names (`mail_autoload.php`):
 - `EzcMailTransportMta` ==> `transports/transport_mta.php`
 - `ezcMailTransportSmtplib` ==> `transports/transport_smtplib.php`
 - `ezcMailException` ==> `exceptions/mail_exception.php`
- Some problems:
 - Clashes in first part of the classname
 - Needs installation into correct place for development

Architecture – Documentation

- Documentation:
 - Makes a library usable
 - API documentation
 - Examples
 - Tutorials
 - Articles
- PHP Documentor:
 - Made for PHP
 - In use by PEAR
 - Supported by PHP IDEs

Architecture - Dependencies

- The less the better
- Only if really necessary
- Dependency-only packages
- Tie-Ins



Example application – Overview

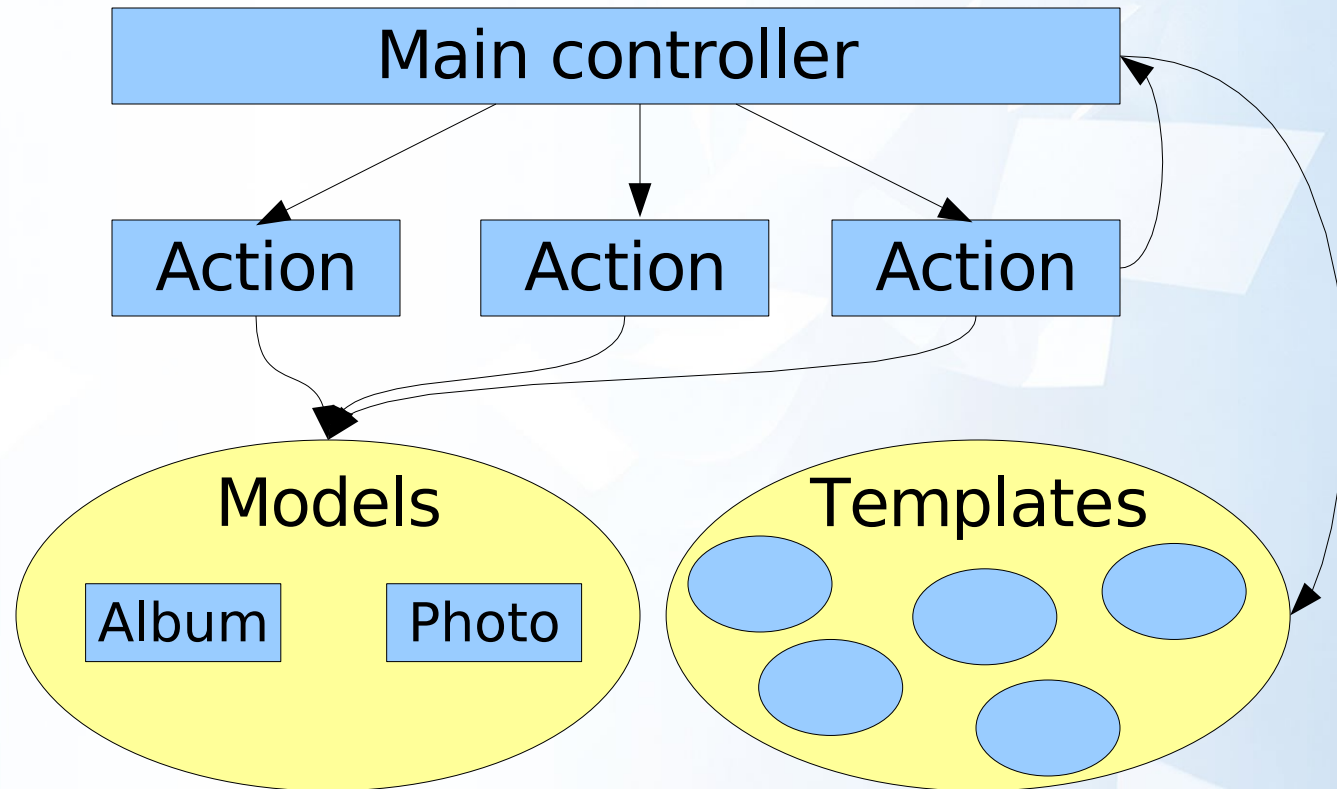
- A Google rank checker
- Uses a MVC pattern
- Provides actions to...
 - ... add a new site
 - ... add a new search
 - ... view all the data
- Uses the following components:
 - Database
 - PersistentObject
 - Config
 - UserInput
 - Template

Example application – Cut down

- We cannot look at the complete application, because we have not enough time... :(
- But I will start digging into it... :)
- If I get toooo much over my time slot, please stop my!
- What we will see is:
 - How a new site is created in the system.
 - How the overview on the site searches is realized.

Example application – General architecture

- MVC (Model-View-Controller) pattern



Open part

- Now it is time for...
 - Questions and answers
 - Feedback
 - Ideas
 - Critics
 - **Or simply for lunch, if you are hungry...**

The bitter end...

- Thanks for listening!
- I hope you...
 - ... learned what you expected
 - ... will try out eZ components
 - ... had a good time (possibly slept well? ;)
- If you want to learn more:
 - The eZ components website
 - http://ez.no/products/ez_components
 - The eZ components documentation
 - <http://ez.no/doc/components/overview/latest>
 - The eZ components forum
 - http://ez.no/community/forum/ez_components
 - The eZ components mailinglist
 - [<components@lists.ez.no>](mailto:components@lists.ez.no)
 - My email address:
 - [<ts@ez.no>](mailto:ts@ez.no)